AUTO CARRIER TRANSPORTER LOADING AND UNLOADING IMPROVEMENT

THESIS

Brian M. Miller, Captain, USAF

AFIT/GAI/ENS/03-02

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

AFIT/GAI/ENS/03-02

AUTO CARRIER TRANSPORTER LOADING AND UNLOADING IMPROVEMENT

THESIS

Presented to the Faculty

Department of Operational Sciences

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Aerospace and Information Operations

Brian M. Miller, BS, MS

Captain, USAF

March 2003

AFIT/GAI/ENS/03-02

AUTO CARRIER TRANSPORTER LOADING AND UNLOADING IMPROVEMENT

Brian M. Miller, BS, MS
Captain, USAF

Approved:

/signed/

Raymond R. Hill, Jr., Lt Col, USAF (RET) date

/signed/

James T. Moore, Lt Col, USAF (RET) date

Acknowledgments

I would like to express my sincere appreciation to my thesis advisor, Dr. Raymond Hill, and Dr James T. Moore my thesis reader, for their guidance and support throughout the course of this thesis effort. The insight and experience was certainly appreciated.

I would also like to thank my wife; she supports me and strengthens me in everything that I do. She is my best friend, without her I am less. I love her deeply, truly and forever.

Brian M. Miller

Table of Contents

Page

Page

List of Figures

Page

List of Tables

AFIT/GAI/ENS/03-02

Abstract

Thousands of vehicles are transported every day from one location to another on Auto Carrier Transports. This process can be represented as a Pickup and Delivery Problem. Solving this instance can lead to the savings of thousands of dollars.

The intent of this research is to develop an algorithm to solve the Auto Carrier Transport Pickup and Delivery Problem. In doing so, the focus is to limit the total times vehicles are placed on and taken off the carrier, otherwise known as the number of loads.

Results show that it is possible to either modify the routing or the position of the vehicles on the carrier to limit the loads. Which process is used depends on the distances the carrier is traveling and how many vehicles are awaiting pickup.

AUTO CARRIER TRANSPORTER LOADING AND UNLOADING IMPROVEMENT

I. Introduction

Introduction

Thousands of vehicles (cars, trucks and vans) are transported every day from one location to another. These vehicles are moved around the country on Auto Carrier Transports (ACTs). Delivering these vehicles involves large quantities of time, money and energy. Any reduction in time and energy results in saving significant amounts of money. Therefore, any improvement towards some form of an optimal solution is valuable.

Tadei, Perboi and Della Croce (2000) proved the Auto Carrier Problem is NP-hard. They draw parallels between a known NP-hard problem, the Multiple Knapsack problem, with a simplified version of the Auto Carrier Problem where all the vehicles originated from a single location and are delivered to a common location. Increasing the number of constraints to include multiple destinations, delivery times and carrier capacity increases the complexity of the problem.

Problem Definition

The problem addressed in this research is a pickup and delivery problem with side constraints on delivery times and carrier capacity. The aim of this problem is to minimize both distance driven and number of loadings. Loadings involve unloading a vehicle from a carrier, loading a vehicle onto a carrier, or reloading a vehicle onto a carrier that was removed to gain access to vehicles deeper on the carrier. Increases in either distance driven or loads result in expenditure of more time and money. Furthermore, increases in loadings increase the risk of damage to vehicles resulting in lower resale values.

The problem can be further broken down into two types: new vehicles delivered to new car dealers and used vehicles picked up and delivered to and from various locations to include purchasers/sellers home, used car dealers, and new car dealers. The delivery of new vehicles to new car dealers is a slightly easier problem for two reasons. First, all the vehicles are picked up at the same location. Secondly, an ACT is filled with vehicles going to one or reasonable co-located new-car dealers.

Routing used, versus new, vehicles is more complicated because the ACTs continue to pickup and deliver throughout the route. Furthermore, routes may change as new pickups and deliveries are added. ACTs also have not only a capacity constraint, but the limitation that certain positions on the ACT have size limitations.

Scope and Methodology

The complete Auto Carrier Problem is, as previously stated, a complex problem. For this initial effort, we simplify the specific problem addressed. The first simplification was to remove the time windows. This was reasonable as we are mainly looking at the total number of vehicle loads. Time windows in this situation will effect the overall routing of the Auto Carrier and can be included in more complex algorithms solving the load problem. The second simplification was to remove the vehicle size limitations. Size limitations determine which vehicles can be assigned to each location on the ACT. The third simplification was to convert the ACT into two flat levels that unload straight. Two flat levels on an ACT simplifies the car loading and unloading process. The final simplification was to number locations and remove the calculations used to determine the distance between the two locations.

We used a combination of a construction heuristic and improvement techniques to develop an algorithm that attempts to optimize the route while minimizing the number of ACT loadings. We wrote the algorithm in Matlab code. We tested it on problems varying the combinations of ACTs, vehicles, locations, maximum distance between locations and percentage of vehicles still requiring pickup.

Overview

Chapter Two presents a brief overview of literature relating to this topic. Chapter Three presents the algorithm developed to solve this problem. Chapter Four presents a comparison of results. Chapter Five presents conclusions and recommendations for future work. Appendix A describes a test problem generator developed for and used in this research.

II. Literature Review

Introduction

This chapter reviews the literature pertinent to the pickup and delivery problem (PDP) of automobiles. Literature on this particular problem is sparse. Thus, to provide an understanding of our particular problem, this chapter first examines the general vehicle routing problem (VRP), then the PDP, and finally the Auto Carrier Transportation (ACT) problem. Carlton (1995) suggested a hierarchical classification scheme for the general VRP. He viewed the hierarchy 'as a three dimensional chess board' with the traveling salesman problem (TSP) as the first level, the VRP on the second level and the PDP on the third level. Moving from the first level to the second adds the constraint of capacity. Moving from the second level to the third level adds the precedence constraint. Each level is divided into five groups based upon the number of vehicles, types of vehicles in the problem (single, multiple homogeneous or multiple non homogenous), the number of depots (one or more), and whether there are constraints on the route length or constraints on the service time.

Vehicle Routing Problem

The VRP is a heavily researched problem with many different solution approaches. Our focus was on work that included time windows, heuristics, and multiple vehicles.

Lee and Ueng (1999) studied how to balance the work load of drivers for a transportation company in Taiwan. Their definition of work load involved the time to complete the route and how much loading or unloading was accomplished at each location. While not a traditional consideration balanced work loads increase driver satisfaction. Although this may increase operational costs, it does provide a long term benefit: increased quality of service.

Lee and Ueng (1999) provide an integer programming model of their problem. The size and complexity of the model made it difficult to solve exactly, so they introduce a nine step heuristic algorithm. Step one determines the minimum number of vehicles needed to meet the demand of the network of locations by dividing the demand by the vehicle capacity. Step two calculates the work time per vehicle by adding the unload time and the travel time. If the work time is greater than the maximum time limit, the number of vehicles required is increased. Step three through five assign two destinations to each vehicle while preventing violation of vehicle load limits and working time limits. Step six determines the present work load for each vehicle. Step seven determines the vehicle with the shortest working time and assigns an unserviced location. Step eight repeats step six and seven until all locations are serviced. Step nine increases the number of vehicles by one if the vehicle load limit or the working time limit are maximized without all locations serviced and returns the algorithm to step two. Their heuristic was able to balance vehicle work time within five percent of the optimal for the eleven test cases. The significant difference was in computer processing time where the optimal method averaged 5643 seconds for the eleven test cases versus 28 seconds for the heuristic.

Solomon and Desrosiers (1988) discussed time window effects on different types of routing problems, such as single and multiple TSPs, the shortest path problem, the minimum spanning tree problem, the generic VRP and the PDP. While they do not offer a process to solve any of these problems, they do reference methods used to solve routing problems with time windows.

Pooley (1992) offered an algorithm for the less-than-truckload vice a multiple-stop truckload VRP. He questioned what to do if a load did not completely fill a truck. Should the truck be sent less than full or should the loads of two or more locations be combined on one truck that has multiple stops? Finally, if multiple loads are combined, which destinations should be combined and which route will the truck take. To solve this problem, he modified the Clarke and Wright savings algorithm. He changed the evaluation criteria from distance to cost, changed the basic savings calculation to recognize different types of vehicles, and finally adjusted the algorithm to recognize the option of switching vehicles. By applying this modified algorithm to test cases with fifteen to forty different destinations, he was able to cut transportation cost by approximately fifteen percent.

Liu and Shen (1999) propose a two stage metaheuristic based on a neighborhood structure for a VRP with time windows. The neighborhood structure used in this case is the relationship between routes and destinations. Their first stage constructs routes by estimating the lower bound of vehicles needed to satisfy unrouted customers and constructing a corresponding set of partial routes. The first stage uses those routes to service customers not served. This continues until there are no customers that can be added while staying in the feasible solution space. This first step is repeated until the total unrouted demand by customers is smaller than a specific threshold. The suggested threshold is either the vehicle capacity, maximal utilization rate times the vehicle capacity, or average utilization rate times the vehicle capacity. This lower bound is generally weak, which is why the second phase sets the number of vehicles available to one less than the answer from phase one. Phase two constructs its routes in parallel repeating the same steps in phase one. Using standardized benchmarked vehicle routing data sets originally created by Solomon, they compared their results with other heuristics. Although their approach did not find best solutions in every situation, it did do well on large-scale problems.

Guan and Zhu (1998) considered vehicle routing where the vehicle can carry multiple objects; referred to as multiple capacity. They include the possibility of *preemptive drops* where a vehicle is unloaded prior to arriving at its destination along with unit or multiple capacity. This allows an item to be routed from its origin to destination using different vehicles. This increases the complexity of the routing while offering overall cost savings. Multiple capacity also creates four kinds of VRP: unit capacity nonpreemptive, unit capacity preemptive, multiple capacity nonpreemptive and multiple capacity preemptive. Nonpreemptive means no objects are unloaded at any location except their final location. Preemptive means objects can be unloaded at intermediate locations. They offer algorithms for each situation that work if each object is moved in the predetermined direction.

Pickup and Delivery

The general pickup and delivery problem (PDP) seeks a set of routes constructed to satisfy requests for picking up and delivering a commodity. Savelsbergh and Sol (1995) describe the general problem and a few of its special cases. They also present a general model designed to handle problems with a single origin and multiple destinations, multiple origins with a single destination, vehicles with different start and stop locations and real time requests for service.

Min (1989) delved into the problem of the multiple VRP with simultaneous delivery and pick-up points. Few studies address combined delivery and pickup problems. Min's method groups customer nodes into a set of clusters in such a way that the total delivery/pickup size of the customers within each cluster does not exceed vehicle capacity. Next, he assigns trucks to the clusters. Finally, he determines the sequence of simultaneous delivery and pickup services over each cluster while not exceeding vehicle space or weight restrictions. The problem he examined involved small items (books) over a small distance (Columbus Ohio Library district). An open question is how well the method generalizes to a large number of locations and trucks over greater distances.

Renaud, Boctor and Ouenniche (2000) developed heuristics for the PDP. Some of the heuristics they considered and modified include the farthest insertion heuristic and the k-exchange improvement heuristic. They proposed a two-phase algorithm. Phase one is a double insertion heuristic that constructs a tour using a local optimizing component that adds each delivery customer with its associated pickup customer. Phase two is a deletion and re-insertion improvement algorithm that uses a 4-Opt improvement heuristic. They finally compared the results of the double insertion heuristic with the deletion and re-insertion improvement algorithm and four other heuristics: the Double Cycle 1, the cheapest insertion, the farthest insertion and Psaraftis algorithm. One of their interesting conclusions was that "important savings may be obtained by visiting pickup and delivery customers alternatively instead of simply visiting all pickup customers before visiting delivery customers."

Time windows define allowable delivery times. Hard time windows force complying with delivery/pickup times. Soft time windows penalize window violations. Dumas, Desrosiers and Soumis (1991) discuss the theory and methodology for the PDP with time windows for a single vehicle and divide the problem into two pieces. The first piece ensures that each transportation request is satisfied exactly once. The second piece determines the shortest path with pickup, delivery and time constraints. Their results suggest that time windows and distribution of load demands have the highest impact both on the solution result and solution time.

Nanry and Barnes (1999) created a reactive tabu search for the pickup and delivery problem with time windows for a vehicle fleet housed at one depot. Their reactive tabu search uses three distinct move neighborhoods: single paired insertion (SPI), swapping pairs between routes (SBR) and within route insertion (WRI). SPI consists of attempts to move a pickup and delivery from its current route to another. The move is permitted if the time window and capacity constraints are not violated. This gives the SPI the possibility of eliminating routes, but only if the new solution is feasible. By eliminating routes, the total number of vehicles is decreased leading to the greatest overall improvement of the solution. SBR is used if the SPI reaches a point where no acceptable move exists. To find new areas of the feasible solution space, SBR swaps a pickup and delivery from one route with a pickup and delivery from a different route. This often results in an infeasible solution, which is fixed by the WRI move. The WRI attempts to move either the pickup or delivery forward or backwards in their assigned route. This effectively reorders the customers to lessen or remove constraint violations or improve feasible solutions. The reactive tabu search starts with the SPI, selecting the most favorable non-tabu move. It then runs WRI until the average time window length is greater than twenty five percent the average route duration length. When there are no feasible WRI moves, SPI moves are accomplished. When there are no feasible SPI moves, SBR are accomplished. When there are no SBR moves, the algorithm uses the best move available.

Nanry and Barnes were able to find optimal solutions to fourteen of fifteen 50-customers problems and eight of nine 100-customers problems of the Solomon's benchmark vehicle routing test set problems with time windows, while running for significantly shorter times. Their algorithm “consistently returned a solution within one percent, on average in a fraction of the computational effort” when compared to best known results.

Fagerholt (2001) describes the PDP as applied to a multi-ship scenario with soft time window constraints. His approach violates some time windows in order to achieve an overall better schedule that significantly reduces transportation cost. He calculates all or a fair number of candidate schedules and their operating and inconvenience cost. Then he represents the candidate schedules as columns of a set partitioning problem.

Auto Carrier Transportation

Tadei, Perobli and Croce (2002) consider the problem of transporting vehicles from a set of delivery ready vehicles to new car auto dealers. This variation of the ACT picks up vehicles at one central location and delivers the vehicles to one or more locations. They provide a detailed look at an Auto Carrier and how it can best be loaded by categorizing vehicles into classes based on the amount of space required to load the car. Each vehicle is also prioritized according to its delivery date with a penalty for late deliveries. Destination locations were grouped together into geographical regions to simplify the calculations by removing unlikely combinations of destinations.

The overall heuristic they developed consisted of three main steps. Step one assigns auto carriers to regions, breaking the general problem into regional subproblems. Step two considers each auto carrier in-turn by building a feasible solution where they add autos to the carrier until it is full. If a dealer’s demand does not fill the carrier, autos requiring delivery to a nearby dealer are added until the carrier is full. Step three attempts to improve the initial solution by using a basic swap improvement search. It compares swaps between loaded vehicles and non-loaded vehicles destined for the same region.

Tadei, et al. (2002) tested their algorithm on both real world and synthetic data sets. The algorithm returned solutions in "several minutes" that had "an average deviation from a simple upper bound lower than three percent."

Agbegha, Ballou and Mathur (1998) consider ACT loading of new vehicles. Their goal was to limit the unloading of vehicles primarily to prevent damage while reducing delivery cost. To accomplish this, they represent the ACT as a directed network where each slot is represented by a node and an arc from node *i* to node *j* means that the car on *i* must be unloaded before the car on node *j* can be unloaded through slot *i*. This creates a loading network for the trailer that can be solved as a minimum-weight spanning tree for each car going to each destination. This can then be used to consider the unloading cost for each destination. Nodes are restricted to the type of vehicles based on their size available for assignment.

Agbegha et al. (1998) note that there are further problems in the ACT beyond the one they worked. One of the problems is "connecting the loading problem to the vehicle-routing aspect of the problem." This is the area the current research investigates.

III. Methodology

Introduction

We developed an algorithm to determine a best ACT route to minimize total load times as well as minimizing the routing time. Our approach combines heuristic techniques and finite algorithms. In this chapter, we describe our approach and how we developed it. By convention, “loading” refers to the initial placing of a vehicle on an ACT at its pickup location, “reloading” refers to removing and replacing vehicles on the ACT to gain access, and “unloading” refers to removing a vehicle at its destination. “Loads” refer to how many times loading, reloading or unloading occurs at a particular location for an ACT. A tour refers to the route, and associated pickup and delivery schedule, for a particular ACT.

Construction Heuristics

Construction heuristics are used to generate an initial solution to a problem. We use a greedy heuristic to generate our initial solution. Our greedy heuristic starts with the present location of the loaded Auto Carrier. For each vehicle on the ACT, the nearest remaining vehicle destination is added to the current route. This continues until all loaded vehicle destinations are considered.

Improvement Heuristics

Improvement heuristics start with some solution and examine changes to the solution in an attempt to improve that solution. Depending on the improvement heuristic, some intermediate solutions do not need to be feasible as infeasible solutions can sometimes lead to good feasible solutions.

One improvement heuristic, the k-opt heuristic, is used in our algorithm. A k-opt heuristic deletes k edges and replaces them by k other edges. After the initial route is created by the construction heuristic, a three-opt heuristic is run on the route. This is followed by a two-opt heuristic. We consider the three-opt to be a coarse improvement as an improvement will be larger than any improvement generated during the two-opt. We consider an improvement generated during the two-opt to be a fine improvement on the route. After all possible changes are examined the lowest total distance tour encountered is selected as the improved tour.

Initial Situation

For our approach, we assume that the Auto Carrier is at some location with a number of preloaded vehicles destined for determined locations. There are also vehicles waiting at locations along or near the carrier's route targeted for pickup and delivery to other destinations. Some of the pickup and dropoff destinations will already be part of the Auto Carrier route. Thus, we consider four different groups of vehicles: those pre-loaded on an ACT, those that can be loaded and delivered based on the Auto Carrier's present route, those that require adding either the pickup or delivery location to the route, and those that require the addition of both the pickup and delivery location to the route. Pre-loaded vehicles are considered by the delivery portion of the algorithm, while the remaining vehicles are considered by the pickup and delivery portion of the algorithm.

Agbegha et al. (1998) describe four different trailer types with different capacities and different loading networks. We, however, considered a simplified carrier design consisting of two flat levels, with ten vehicle positions. Figure 1 shows an example of our notional carrier. To off-load vehicle positions six through ten, position one must empty (temporarily off-load any vehicle in position one).

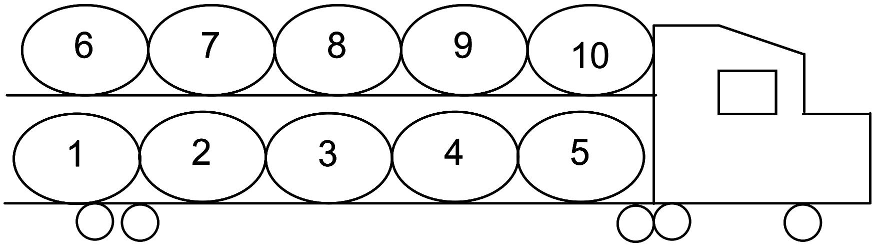

Figure 1. Carrier Vehicle Position Numbers

As the carrier progresses along a route, the number of loaded vehicles and their position on the ACT may change. We therefore developed a data structure that tracked the vehicle number, the vehicle destination location, the carrier route and the number of loads at each location. This data is tracked in the ACT Matrix, referred to as the ACTMat. The change in the positions of the vehicle numbers and their corresponding destinations are tracked in two variables, VdateMat and UpdateMat, respectively. The number of vehicles on the ACT after each location is stored in VplMat.

Consider a worst case scenario. This occurs when vehicles are loaded opposite of their delivery order thereby maximizing reloading at any delivery location. In this instance, if the destinations are opposite the position numbers (i.e. vehicle number ten to destination one, vehicle number nine to destination two, etc.), a full load of ten vehicles, with one vehicle dropped off at each location results in sixty loading actions, as shown in Table 1. This assumes that no additional vehicles are added enroute. A solution is to completely off-load every vehicle and reload the vehicles in reverse order so that the destinations and positions are matched. The result of this action would be thirty loads.

Table 1. Worst Case Loading Scenario

Position number	Destination	Off loads	Reloads	Total Loads
10	1	1	10	11
9	2	1	8	9
8	3	1	6	7
7	4	1	4	5
6	5	1	2	3
5	6	1	8	9
4	7	1	6	7
3	8	1	4	5
2	9	1	2	3
1	10	1	0	1
				60

The best case situation would be to start off with the destinations and load positions in agreement (i.e. vehicle number one to destination one, vehicle number two to destination two, etc.). This would result in ten loads as shown in Table 2.

While both cases are possible, their likelihood is small. Adding additional vehicles to include pickups further reduces their chance of occurrence. These cases, however, provide simple bounds to the loading effort on a route.

Table 2. Best Case Loading Scenario

Position number	Destination	Off loads	Reloads	Total Loads
10	10	1	0	1
9	9	1	0	1
8	8	1	0	1
7	7	1	0	1
6	6	1	0	1
5	5	1	0	1
4	4	1	0	1
3	3	1	0	1
2	2	1	0	1
1	1	1	0	1
				10

To balance distance traveled and loading effort, our algorithm considers any vehicles requiring loading, unloading and reloading on to the carrier. The vehicles are reloaded onto the carrier last location (on the route) first and nearest location (on the route) last. This orders the vehicles so that fewer vehicles must be temporarily removed from the ACT to gain access to vehicles being delivered to the current location. This approach takes advantage of any instance where reloaded vehicles positions can be adjusted to better match the routing. This prevents us from ever getting close to the worst case scenario or completely reloading all the vehicles at every destination in the route. It also allows for vehicles to be added during the route without completely disrupting the off-loading of vehicles.

If the routes involve long distances between locations, driving time will dominate the total time. When driving time dominates, it is advantageous to minimize distance and increase reloads to re-organize the ACT. If routes involve shorter distances between locations, the loading time dominates the total time. When this occurs, it is advantageous to route the ACT to accommodate the loading pattern and their deliveries to minimize unloading actions.

Delivery

Vehicle delivery routing must balance distance traveled by the ACT and total number of loads accrued during the route. The steps required are initial route, improved route, and balancing loading and distance. Administratively, we track the number of vehicles on the carrier and their positions as they are unloaded and reloaded.

The initial route was generated using the previously defined greedy construction heuristic. This route was improved using a three-opt, and then a two-opt move. The three-opt move takes a location in the route and inserts it between two other locations in the route. The two-opt swaps the order of pairs of locations in the route. Possible moves are considered and the route with the most improvement, if any, becomes the new route.

Once a final route is determined, we calculate the number of loads and the total time to travel the route. This is done without changing vehicle positions. Naturally some vehicles must be reloaded to access vehicles targeted for delivery. We assumed loading time as fifteen minutes and driving speed as fifty-five miles per hour regardless of roads used. The total combined time is stored as the initial solution time.

The last step in the delivery algorithm re-evaluates the route comparing travel time and loading time. This is accomplished by starting at the initial location and determining which destination has the lowest combined travel time and load time. The location with the lowest combined time is selected as the next location in the route. This is accomplished until there are no remaining locations. The total time of this new route is calculated and stored as the improved time.

An example of these last steps (Figure 2) starts with the original route at location one, then proceeds in order to locations two and three. The time to travel from location one to location two is twenty minutes, from location two to location three is thirty minutes and from location one to location three is thirty minutes. The load time for location two, when it follows location one, is forty-five minutes. The load time for location three, when it follows location one is fifteen minutes. We would then select location three as the next destination after location one because the total time for location three would be forty-five minutes compared with the sixty-five minutes for location two. The load time at location two now becomes fifteen minutes, with a total time for location two as forty-five minutes. For the entire route, the total time is ninety minutes. This compares with the initial solution of one hundred and ten minutes.

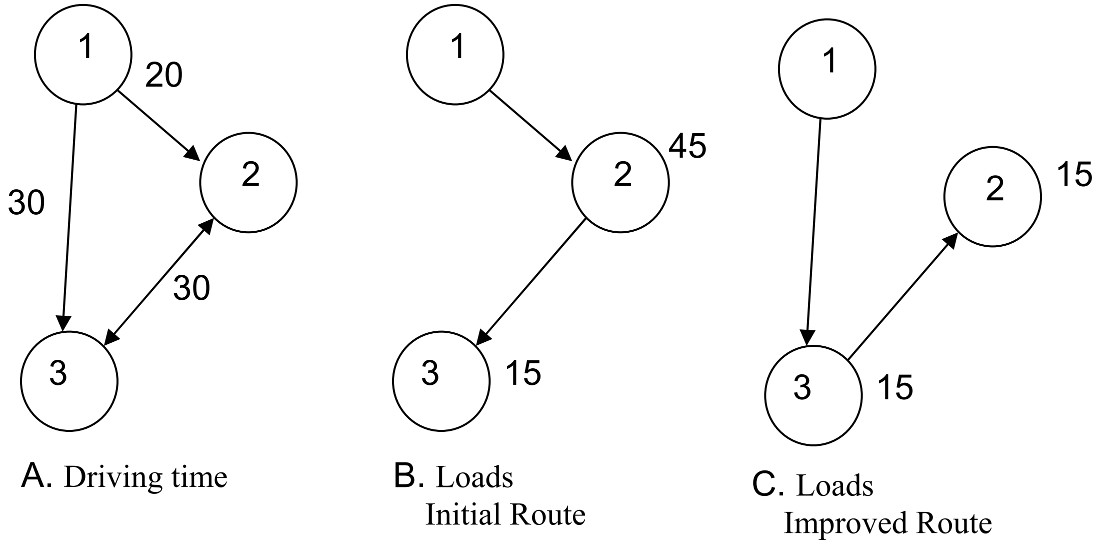

Figure 2. Delivery Route Improvement

The initial solution time and the improved time are compared, and the lower time is selected as the delivery time. The route that generated the lower delivery time is now considered the delivery route. The number of vehicles on the carrier at each location is the calculated for use in the next section.

Pickup and Delivery

Pickup and delivery considerations change the problem significantly by adding vehicles and in most cases locations to the present route. The easy case involves a vehicle picked up and dropped off along the carrier's route. A more complex situation involves adding one additional location to the route. The most complex situation is involves adding both the pickup and delivery location to the route. The steps required are vehicle determination, location insertion, and balancing loading and distance. Administratively, we track the number of vehicles on the carrier and their positions as they are loaded, unloaded and reloaded.

The first step in the pickup portion of the algorithm determined the vehicles that required pickup, their pickup locations, and their destination locations. A list of distances from carrier route destinations to pickup locations was created. Where pickup and delivery locations occurred during the route, the distance was zero so adding the vehicle added no distance to the route. Where either the pickup location, the delivery location or both locations was not on the route, the least increase in carrier travel distance was sought. This approach determined where to insert any new locations in the route. Once in the route, carrier and route data structure were updated. This process was then repeated until all the pickup and destination locations required were added to the routes of the carriers, subject to carrier vehicle limits.

Since pickup and delivery is a dynamic situation, the list of vehicles was again compared with the updated carrier routes to generate a matrix tracking where vehicles were loaded. This data is used to determine which vehicles are added to the carrier as the carriers service each location, unloading any vehicles destined for that location. At locations where either no vehicles are being unloaded or there are more vehicles being loaded than unloaded an imbalance between positions available to the algorithm and vehicles awaiting loading occurs. To handle the imbalance, phantom vehicles are inserted on the carrier. The number of phantom vehicles added is equal to the imbalance. These phantom vehicles are assigned to earliest empty positions with a destination of the location with the imbalance. The algorithm removes all the vehicles up to and including the phantom vehicle(s) from the carrier creating a space for the vehicle(s) to be loaded. Prior to any vehicles being reloaded onto the carrier, they are again sorted by their destinations. The destinations that occur later in the route are again loaded into higher position numbers.

Once positions have been calculated for all the vehicles during the route, the total number of vehicles on the carrier is recalculated. This information is then used to determine if there are any locations that do not result in a load or unload. This can occur when a location already on the route is forced to be added earlier in the route due to a pickup needing to be before a certain destination. Providing all other actions required at that location can be performed without overflowing the carrier’s capacity, the original visit to that location now becomes superfluous. Any superfluous locations are removed.

The pickup and delivery time can now be calculated by determining the total travel time from the total distance traveled and the total load time from the total number of loads accomplished.

Summary

We have developed an algorithm that takes advantage of the nature of the problem. Instead of attempting to completely organize the ACT at each destination, we have tried to use rerouting and reloading to minimize total time and loads. This gives us the ability to re-route the carrier as additional vehicles are added to the problems, even if they have new pickup or destination locations.

IV. Results and Analysis

Introduction

This chapter presents the results of testing the algorithm. A full solution is presented in depth to detail the algorithm. This is followed by an empirical study of twelve pickup and delivery variations designed to explore the overall quality of the algorithm. Finally, we consider the overall effectiveness of the algorithm to ACT routing and scheduling.

Data Format

A simple data display, is used to depict the details of the algorithm. The data displays the relative position of the vehicles on the ACT (Figure 3) using the format in Table 3. Note the correspondence between vehicle numbers. Blanks indicate no vehicle. If there are five vehicles presently on the carrier, vehicle numbers 3, 16, 21, 25, and 30, in positions 2, 3, 6, 7, 8, respectively, the carrier vehicle numbers are shown in Table 4. If those same vehicles have destinations of 10, 4, 9, 20, and 26, respectively, the destinations would be displayed as they are in Table 5.

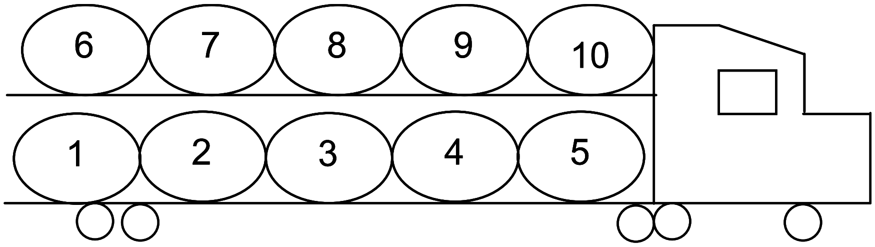

Figure 3. Carrier Vehicle Position Numbers

Table 3. Vehicle Positions Numbers

Level 2	6	7	8	9	10
Level 1	1	2	3	4	5

Table 4. Example of Vehicles on Carrier

ACT number	Vehicle numbers				
1	21	25	30		
		3	16		

Table 5. Example of Vehicle Destinations

ACT number	Vehicle destinations				
1	9	20	26		
		10	4		

First Data Run

One run of the algorithm is detailed to show the basic data structure and improvement approaches. This trial used three ACTs, thirty vehicles, and twenty locations. Of the thirty vehicles, eight were awaiting pickup. The vehicle numbers and their positions on the ACTs are displayed in Table 6. The corresponding vehicle destinations are provided in Table 7. Table 8 shows the vehicle numbers awaiting pickup and their pickup and destination locations.

Table 6. First Data Run Carrier Vehicle Numbers

ACT	Vehicle numbers				
1	14	16	18	21	
	1	2	5	11	13
2					
	3	9	17	22	
3	19	20			
	4	6	8	10	12

Table 7. First Data Run Carrier Vehicle Destinations

ACT	Vehicle destinations				
1	12	16	12	5	
	9	20	6	8	4
2					
	1	2	17	2	
3	6	3			
	18	3	5	12	8

Table 8. First Data Run Vehicles Awaiting Pickup

Vehicle number	Pickup location	Delivery location
23	6	20
24	5	10
25	6	14
26	20	16
27	14	3
28	2	10
29	6	17
30	20	1

The delivery portion of the algorithm calculates the best route. Reloading and unloading are based on the vehicles presently on the carrier. The routes are listed in Table 9 starting with the present location of the ACTs. Table 10 shows the number of vehicles on the carriers after each location. The carriers' configuration changes through the fourth destination are shown in Table 11. Table 11, ACT 1, shows how the algorithm reloads vehicles two and five to facilitate delivery later in the route. Similarly, ACT 3 reloads vehicle four, six, eight and ten differently. Table 12 shows the number of loads at each location.

Table 9. First Data Run Delivery Only Routes

ACT	Vehicle destinations in delivery order								
1	11	8	12	4	9	5	6	20	16
2	19	17	1	2					
3	15	8	12	3	5	18	6		

Table 10. First Data Run Delivery Only Vehicle Quantities

ACT	Number of vehicles on ACT after each location								
1	9	8	6	5	4	3	2	1	0
2	4	3	2	0					
3	7	6	5	3	2	1	0		

Table 11. First Data Run Delivery Only Carrier Status During Route

Location	ACT	ACT position and vehicle number in that position				
Origin	1	14	16	18	21	
		1	2	5	11	13
	2					
		3	9	17	22	
	3	19	20			
		4	6	8	10	12
1st destination	1	14	16	18	21	
		1	5	2		13
	2					
		3	9		22	
	3	19	20			
		10	6	8	4	
2nd destination	1		16		21	
		1	5	2		13
	2					
			9		22	
	3	19	20			
			6	8	4	

3rd destination	1		16		21	
		1	5	2		
	2					
	3	19				
				8	4	
4th destination	1		16		21	
			5	2		
	2					
	3	19				
					4	

Table 12. First Data Run Delivery Only Carrier Loads per Location

ACT	Loads per location								Total Loads
1	7	6	7	1	3	1	1	1	27
2	5	1	2						8
3	9	1	4	1	1	1			17

The final portion of the algorithm determines which vehicles require pickup and any locations that must be added into the route. These locations are inserted in a route to minimize change to the overall route length, subject to carrier capacity. The vehicles awaiting pickup, the carriers that will pick them up, and their pickup and delivery locations are shown in Table 13. The revised routes and the total number of vehicles at each location are displayed in Tables 14 and 15, respectively. The carriers' configuration changes through the fourth destination are shown in Table 16. Table 14 shows how the new locations have been added into the carriers' routes. Location 20 has been added earlier in ACT 1's route to facilitate adding vehicles twenty-six and thirty prior to their destinations of sixteen and one, respectively. This allows vehicle two to be unloaded at the fourth location in the route while vehicles twenty-six and thirty are added. These changes are shown in Table 16. Under normal circumstances, the original visit to location twenty would now be removed except that vehicle twenty-six, added at location six, is destine for location twenty. Table 17 shows the number of loads at each location.

Table 13. First Data Run Vehicles Awaiting Pickup Carrier Assignment

Vehicle number	ACT scheduled	Pickup location	Delivery location
23	1	6	20
24	3	5	10
25	1	6	14
26	1	20	16
27	1	14	3
28	2	2	10
29	1	6	17
30	1	20	1

Table 14. First Data Run Pickup and Delivery Routes

ACT		Vehicle destinations in delivery order													
1	Delivery	11	8	12	4	9	5	6	20	16					
	Pickup & Delivery	11	8	12	4	20	9	5	1	6	14	17	3	20	16
2	Delivery	19	17	1	2										
	Pickup & Delivery	19	17	1	2	10									
3	Delivery	15	8	12	3	5	18	6							
	Pickup & Delivery	15	8	12	3	5	18	10	6						

Table 15. First Data Run Pickup and Delivery Vehicle Quantities

ACT		Number of vehicles on ACT at after location													
1	Delivery	9	8	6	5	4	3	2	1	0					
	Pickup & Delivery	9	8	6	5	6	5	4	3	5	5	4	3	2	0
2	Delivery	4	3	2	0										
	Pickup & Delivery	4	3	2	1	0									
3	Delivery	7	6	5	3	2	1	0							
	Pickup & Delivery	7	6	5	3	3	2	1	0						

Table 16. First Data Run Pickup and Delivery Carrier Status During Route

Location	ACT	ACT position and vehicle number in that position				
Origin	1	14	16	18	21	
		1	2	5	11	13
	2					
		3	9	17	22	
	3	19	20			
		4	6	8	10	12
1st destination	1	14	16	18	21	
		2	1	5		13
	2					
		3	9		22	
	3	19	20			
		10	6	8	4	
2nd destination	1		16		21	
		2	1	5		13
	2					
			9		22	
	3	19	20			
			6	8	4	
3rd destination	1		16		21	
		2	1	5		
	2					
		28				
	3					
		19		8	4	
4th destination	1	16	26		21	
		30	1	5		
	3					
		24	19		4	

Table 17. First Data Run Pickup and Delivery Carrier Loads per Location

ACT		Loads per location													Total Loads
1	Delivery	7	6	7	1	3	1	1	1						27
	Pickup & Delivery	7	6	7	5	3	7	1	10	2	3	1	1	2	55
2	Delivery	5	1	2											8
	Pickup & Delivery	5	1	3	1										10
3	Delivery	9	1	4	1	1	1								17
	Pickup & Delivery	9	1	4	4	5	1	1							25

Table 18. First Data Run Times

ACT	Original Time	Delivery Time	Pickup & Delivery time
1	1204.1	1174.1	2776.6
2	812.7	812.7	850.3
3	695.2	545.2	1180.1
Total time	2712.0	2532.0	4807.0

Table 18 is a summation of times, in minutes, for the initial, the delivery and the pickup and delivery routes. The time for the three carriers initially totaled 2712 minutes. After the delivery portion of the algorithm, the total time decreased to 2532 minutes. The final time, which includes pickup and delivery, is 4807 minutes. These assume the previously stated times of fifteen minutes per load and fifty-five miles an hour travel speed. The increase in time can be split into 1705 minutes used to travel an additional 1563 miles to the seven added locations and 570 minutes for thirty-eight additional loads.

Thirty-eight loads might seem like a large increase for adding only eight vehicles, but it does not increase the total loads per vehicle dramatically. Direct comparison between the total delivery loads and the total pickup and delivery loads can not be accomplished because the delivery will not include loads resulting from adding vehicles to a carrier. Therefore only the average number of access loads per vehicle will be considered. For the delivery only portion, carrier one has an average of 2.0 access loads per vehicle. For the pickup and delivery portion, carrier one has an average of 2.27 access loads per vehicle. The carriers two and three start off with average loads per vehicle of 1.0 and 1.43, respectively. After adding the pickups, carrier two has an average of 0.8 loads per vehicle and carrier three has an average of 2.0 loads per vehicle. Comparing across the entire problem the delivery average is 1.6 access loads per vehicle and the pickup and delivery average is 1.93 access loads per vehicle. While the average number of loads per vehicle increases, the increase is not much and results from an increased number of vehicles on the carriers.

Group Data Run

A designed experiment involving random test problems was used to test the algorithm. The variables varied were the percent of total vehicles pre-loaded on the carrier, the number of locations and the maximum distance (MaxDis). The values used are displayed in Table 19. The resulting times are displayed in Table 20 in minutes. Appendix A contains the details of how these parameters were used to generate test problems.

In all cases, the delivery-only algorithm improved the delivery time for each carrier over the route from the construction heuristic. The improvement ranged anywhere from 16 minutes to 132 minutes. When there were vehicles requiring pickup, the delivery time increased due mostly to time spent re-loading vehicles.

Table 19. Data Run Values

Run	Number of ACTs	Number of vehicles	Percent awaiting pickup	Number of Locations	MaxDis
1	3	30	75	15	500
2	3	30	50	15	500
3	3	30	0	15	500
4	3	30	75	30	500
5	3	30	50	30	500
6	3	30	0	30	500
7	3	30	75	15	100
8	3	30	50	15	100
9	3	30	0	15	100
10	3	30	75	30	100
11	3	30	50	30	100
12	3	30	0	30	100

Table 20. Data Run Average Times

Run	Average Carrier Original Time	Average Carrier Delivery Time	Average Carrier Pickup & Delivery Time
1	844	828	2519
2	1178	1108	1985
3	1743	1651	1651
4	743	724	2735
5	1158	1107	2391
6	1739	1640	1640
7	258	238	787
8	402	360	692
9	674	583	583
10	206	188	891
11	372	316	801
12	728	596	596

It was expected that the average distance would increase as locations were added to the route. The reason is because the new locations had to occur either after, in the case of new destinations, or before, in the case of new pickups, certain places in the route. This could lead to the suboptimal insertion of new locations. The worst increase in distance was run two where the average distance traveled between locations increased by 12.07 miles. The best case was run five where the average distance travel between locations decreased by 4.79 miles. The average distances in miles are shown in Table 21.

Table 21. Data Run Distances

Run	Number of Locations	Carrier Average Delivery Distance	Carrier Average Pickup & Delivery Distance	Difference
1	15	222.17	226.35	4.18
2	15	185.88	197.96	12.07
3	15	170.12	170.12	0.00
4	30	198.04	193.83	-4.21
5	30	187.60	182.81	-4.79
6	30	138.47	138.47	0.00
7	15	42.67	45.69	3.02
8	15	38.01	44.95	6.94
9	15	32.50	32.50	0.00
10	30	40.32	37.77	-2.55
11	30	35.71	38.72	3.01
12	30	27.82	27.82	0.00

The dramatic increase in time for the pickup problem is due to the additional loading and re-loading of vehicles. This time increase is pronounced when a large percentage of vehicles are designated for pickup. When few vehicles need pickup, it is easier to find carrier spaces without removing multiple vehicles. When many vehicles await pickup, the carrier is reloaded multiple times to gain access to any open spots on the carrier.

Comparing the decrease in number of locations shows an increase in the total time. This is the result of two opposite reactions. The average distance between locations actually goes down as the number of locations increases and the number of loads goes down as the number of locations decreases. This can be shown using runs three and six which are identical except for the number of locations. Run six's average distance traveled is 32 miles less than run three. This should result in a 35 minute time difference. Yet there is only an 11 minute time difference. Fewer locations allow multiple vehicles to be unloaded and/or loaded at one location lowering the total number of loads.

Increasing the maximum distance between two locations did not appear to have any rational impact. Two standardized distance matrices were used, one for runs one through six and one for seven through twelve. Both matrices had thirty locations randomly created with only the first fifteen used for runs one through three and seven through nine. The locations for the 100 and 500 mile matrices are shown in Figures 4 and 5, respectively. The average distance between the locations for the 100 mile matrix is 54.4 miles and 51.1 miles for the first 15 locations and all 30 locations, respectively. The average distance between the locations for the 500 mile matrix is 259.3 miles and 250.6 miles for the first 15 locations and all 30 locations, respectively.

The difference in the distance had an effect on the improvements generated by the algorithm. We reason that when the average distance between locations are greater, the algorithm generates time savings mostly from improving loading and occasionally improving travel time. Oppositely, where the average distance between locations is smaller, the algorithm forces the carrier to travel to locations that have fewer off-loads.

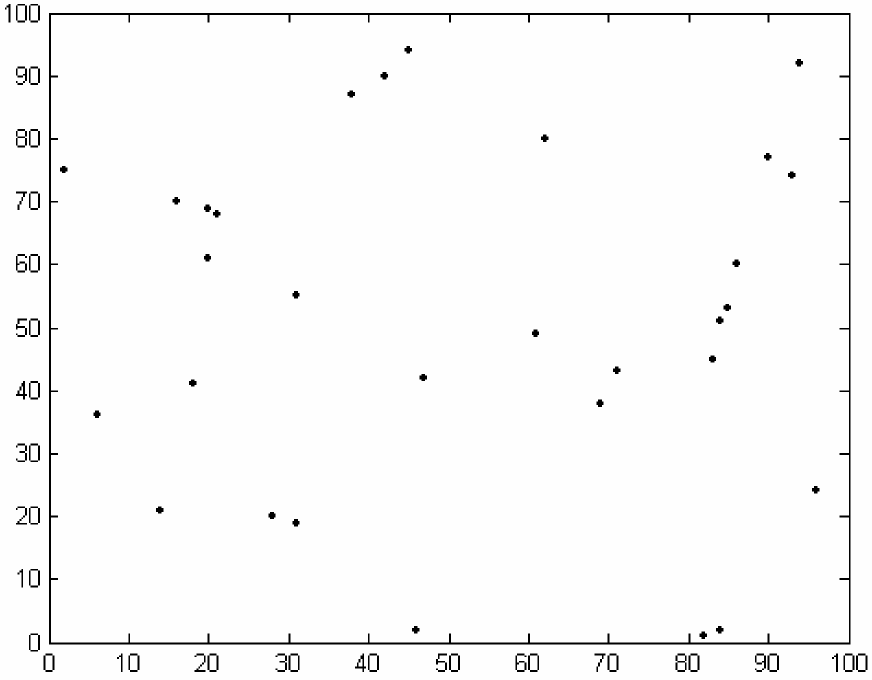

Figure 4. 100 Maximum Distance Locations

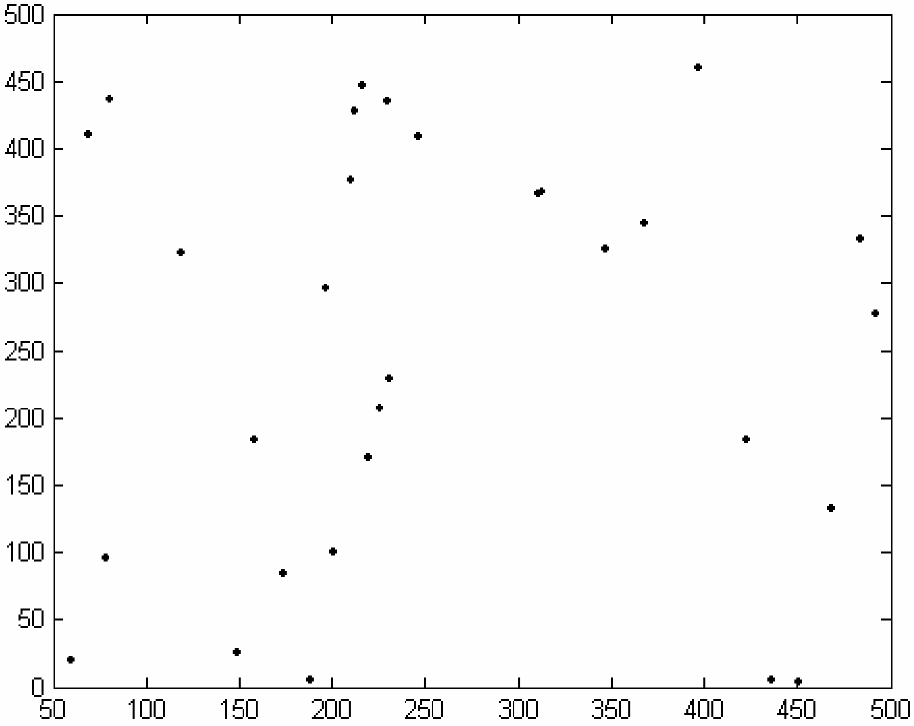

Figure 5. 500 Maximum Distance Locations

Computation Time

The algorithm was implemented in Matlab and run on a Pentium III 700 MHz, 256 Mb, with Windows 2000 Professional. Our average time for each instance was less than seven seconds.

Summary

We attempted to minimize the total number of loads accomplished through rerouting and vehicle shifting. The algorithm developed works well, reducing either the number of loads or the distance driven to provide the minimum time. The tests we ran validate those improvements.

V. Discussion

Introduction

This chapter summaries the research and presents suggestions for future areas of study. It also includes some suggestions about algorithm improvements.

Research Results

This research developed an algorithm for the pickup and delivery problem for Auto Carrier Transports. Numerous pickup and delivery algorithms have been developed for different items, but none for Auto Carrier Transports. The need to not only track the capacity, but to limit the total number of loads accomplished resulted in our developing a completely new algorithm.

Our algorithm solves the delivery of the vehicles presently on the carrier to generate an initial feasible solution. This is accomplished by first determining the best possible route for the carrier to travel. After determining the number of loads based on this route, we attempt to improve the initial values by adjusting vehicle placement to improve any loading or unloading accomplished.

We then determine which vehicles require picking up and which locations need to be added to a route. These locations are compared with locations already serviced to determine how best to minimize the added travel distance. We then analyze the carrier's route to find improvements by adjusting vehicle placement during any loading or unloading to improve total time.

Recommendations for Future Research

We solved the basic problem of the pickup and delivery but there remains more work before the entire problem is solved. For instance, inclusion of time windows, vehicle sizing and carrier networking constraints still remain.

There were indications from the results that we obtained that adjusting the placement of vehicles loaded during the route as pickups might result in additional load savings. Furthermore, there may be additional reduction in loads by determining where large quantities of vehicles are reloaded to determine if any vehicles still on the carrier can be included in the reload if their destination occurs sooner in the route than those presently being reloaded.

Appendix

To test our algorithm, we needed to generate random data sets. We began with the assumption that whenever the algorithm is run, there will be vehicles on the ACTs and there may be vehicles awaiting pickup.

We first randomly chose a location for each ACT. Then for each vehicle to be loaded on an ACT, we randomly chose an ACT on which to place it. If the ACT was full, we re-drew another random choice. Providing there was space, we randomly chose a destination. We then stored the vehicle number and destination in the ACTMat (the matrix that tracks the ACT status). We also stored the ACT that the vehicle was on and the destination in a vehicle matrix (VehMat).

For vehicles not pre-loaded onto an ACT, we randomly generated both a pickup and destination location. They were stored in the VehMat after ensuring that the two locations were not the same.

Bibliography

1. Agbegha, Gerald Y., Ronald H. Ballou and Kamlesh Mathur. "Optimizing Auto-Carrier Loading," *Transportation Science*, Vol 32 No 2: 174-188 (May 1998).

2. Carlton, William B. *A tabu search approach to the general vehicle routing problem*. PhD dissertation. University of Texas at Austin, Austin TX, 1995.

3. Chao, I-Ming. "A tabu search method for the truck and trailer routing problem," *Computers and Operations Research*, 29: 33-51 (2002).

4. Dumas, Yvan, Jacques Desrosiers and Francois Soumis. "The pickup and delivery problem with time windows," *European Journal of Operations Research*, 54: 7-22 (1991).

5. Fagerholt, Kjetil. "Ship scheduling with soft time windows: An optimization based approach," *European Journal of Operations Research*, 131: 559-571: (2001).

6. Guan, D. J. and Xuding Zhu. "Multiple capacity vehicle routing on paths," *SIAM J Discrete Math,* Vol 11 Number 4: 590-602 (November 1998).

7. Lee, Tzong-Ru and Ji-Hwa Ueng. "A study of vehicle routing problems with load-balancing," *International Journal of Physical Distribution and Logistics Management,* 29: 646-658 (1999).

8. Liu, Fuh-Hwa Franklin and Sheng-Yuan Shen. "A route- neighborhood-based metaheuristic for vehicle routing problem with time windows," *European Journal of Operations Research*, 118: 485-504 (1999).

9. Min, Hokey. "The Multiple Vehicle Routing Problem with Simultaneous Delivery Points and Pick-up Points," *Transportation Research Part A*, 5: 377-386 (September 1989).

10. Nanry, William P. and J. Wesley Barnes. "Solving the pickup and delivery problem with time windows using reactive tabu search," *Transportation Research Part B*, 34: 107-121 (2000).

11. Pooley, John. "A vehicle routing problem for the less-than-truckload verse multiple-stop truckload problem," *Journal of Business Logistics,* Vol 13 Is 1: 239-258 (1992).

12. Reeves, Colin R. *Modern Heuristic Techniques for Combinatorial Problems*. Berkshire UK: McGraw-Hill Book Company Europe, 1995.

13. Renaud, Jacquesn, Fayez F. Boctor and Jamal Ouenniche. "A heuristic for the pickup and delivery traveling salesman problem," *Computers and Operations Research*, 27: 905-916 (2000).

14. Savelsbergh, M. W. P. and M. Sol. "The general pickup and delivery problem," *Transportation Science*, Vol 29 No 1: 17-29 (Feburary 1995).

15. Solomon, Marius M. and Jacques Desrosiers. "Time Window Constrained Routing and Scheduling Problems," *Transportation Science*, Vol 22 No 1: 1-14 (Feburary 1988).

16. Tadei, R., G. Perboli and F. Della Croce. "A heuristic algorithm for the auto-carrier transportation problem," *Transportation Science*, Vol 36 No 1: 55-62 (Feburary 2002).

Vita

Captain Brian M. Miller graduated from Farmingdale High School in Farmingdale, New York. After two years at State University of New York (SUNY) at Stony Brook, he enlisted in the United States Air Force. After completing technical training, he was stationed at Griffiss AFB, New York. While stationed at Griffiss, he completed his undergraduate studies at SUNY Institute of Technology at Utica, where he graduated Magnum Cum Laude with a Bachelor of Science degree in Electrical Engineering Technology in May 1992. He was then stationed at RAF Mildenhall, England, where he applied and was accepted to Officer Training School. He was commissioned November 1994 at Maxwell AFB.

He attended Undergraduate Missile training at Vandenberg AFB, California, where he was a distinguished graduate. His next assignment was at Grand Forks AFB, North Dakota as an Intercontinental Ballistic Missile Launch Officer. In April 1998, he was assigned to the 30th Range Squadron, Vandenberg AFB, California where he served as a Range Control Officer, Range Operations Officer, Training Instructor and Chief of Range Training. In May 2002, he entered the Graduate School of Engineering and Management, Air Force Institute of Technology. Upon graduation, he will be assigned to the National Air Intelligence Center, Wright-Patterson AFB.

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 074-0188*

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)* 25-03-2003	**2. REPORT TYPE** **Master's Thesis**	**3. DATES COVERED** *(From – To)* Sep 2002 - Mar 2003

4. TITLE AND SUBTITLE AUTO CARRIER TRANSPORTER LOADING AND UNLOADING IMPROVEMENT	**5a. CONTRACT NUMBER** **5b. GRANT NUMBER** **5c. PROGRAM ELEMENT NUMBER**
6. AUTHOR(S) Miller, Brian, M., Captain, USAF	**5d. PROJECT NUMBER** **5e. TASK NUMBER** **5f. WORK UNIT NUMBER**
7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S) Air Force Institute of Technology Graduate School of Engineering and Management (AFIT/EN) 2950 P Street, Building 640 WPAFB OH 45433-7765	**8. PERFORMING ORGANIZATION REPORT NUMBER** AFIT/GAI/ENS/03-02
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) AFIT/ENS	**10. SPONSOR/MONITOR'S ACRONYM(S)** **11. SPONSOR/MONITOR'S REPORT NUMBER(S)**

12. DISTRIBUTION/AVAILABILITY STATEMENT
APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Thousands of vehicles are transported every day from one location to another on Auto Carrier Transports. This process can be represented as a Pickup and Delivery Problem. Solving this instance can lead to the savings of thousands of dollars.

The intent of this research is to develop an algorithm to solve the Auto Carrier Transport Pickup and Delivery Problem. In doing so, the main goal is to limit the total times vehicles are placed on and taken off the carrier, otherwise known as the number of loads.

Results show that it is possible to either modify the routing or the position of the vehicles on the carrier to limit the loads. Which process is used depends on the distances the carrier is traveling, how many vehicles are awaiting pickup.

15. SUBJECT TERMS
Pickup and Delivery, Auto Carrier Transport

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Raymond R. Hill, Lt Col, USAF (RET) (ENS)
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)* (937) 255-6565, ext 4323; e-mail: Ray.Hill@afit.edu
U	U	U	UU	53	